# CLIMATE CONNECT CASE STUDY

# (SYNOPSIS)

## Problem Statement :

Given Problem Statement includes three .csv files:

- Power_actual
- Weather_actuals
- Weather_forecasting

involving details about a power station and on the basis of that we are supposed to predict the generation of power of the power station.

## Solution Summary :

Started by reading all the .csv files to understand what all is provided to us. On examining the 3 three files power_actual and weather_actual files were to be used for training purpose and weather_forecasting to be used as test set .

Thus, joining power_actual and weather_actual .csv files on '*datetime_loacl'* as common attribute for both the .csv files .

Initially, thought that we have to do Time series forecasting on the given data but after several hours of trials and errors ,I concluded that we can go ahead with it as a Regression problem.

Dropping the unnecessary columns like ghi, gti etc . After, checking df.describe() function it was observed that many attributes were having -9999.0 and -9999 as minimum value. Then tried to identify all columns where these minimum values were present also checked for unique values present in various columns and observed them carefully as there were various columns which involved only NaN and -9999 these two values. Thus, removed those columns as well as it will not add any value to our purpose.

Also, replacing the -9999 values with mean and forward fill method of fillna() function at their respective columns. Again, checking for any presence of missing value just to double sure that there's no missing value left. Encoding the category attributes using LabelEncoder which transformed our data to be completely numeric in all aspects. Checking if there is any multicollinearity in our datset. It was found that 2 columns were having high multicollinearity.Multicollinearity can be treated by two methods either by dropping the columns or by applying PCA, Partial least Square regression. I decided to drop them. After, which outliers were removed using Inter Quartile Range function.

And,after applying StandardScaler on the dataset it was ready to be given to the model. I chose to try with Random Forest model as it has been proven that Random Forest does not make any assumption about the distribution of our data. Random forest gave testing R2 score of approx. 75% which seemed to be quite okay. Additionally, I tried with a library named as LazyPredict which on given training and test data performs numerous Regression and classification models within a few lines of code, which is very much cool. We can compare output of various models in a few lines of code.

## Future Scope:

Given more amount of data may improve the R2 score of the model. Also, by tuning the hyperparameters of RndomForest model R2 score may improve.